CMPE 124 Lab 2: Verifying Theorems with Logic Circuits

Leonardo Blas

***Abstract*—The purpose of this lab project is to verify digital logic theorems by implementing logic circuits.**

I. Introduction

This lab project is based on the implementation of a clock-counter circuit, a logical circuit that tracks time by using bits.
We use this clock's input to verify digital logic theorems, using AND, OR, and NOT gates.

II. Design Methodology

For this lab project, we use the 74LS163 4-bit counter. This 4-bits counter requires not only a voltage source—to function—and a clock—to provide it with time input—but also a binary switch, to reset it and ensure the wave functions it outputs are correct. Additionally, although we use a resistor, its value can be changed or ignored; our 1kΩ resistor is part of the circuit to demonstrate that we are not focusing on the input voltage, but on the H and L states instead.
Using this clock's output, as well as AND, OR, and NOT gates, we plan to verify the following equations:

1. $g_1 = x + x'$, $g_2 = 1$ — $q_0 = x$
2. $g_1 = x * x'$, $g_2 = 0$ — $q_0 = x$
3. $g_1 = x + 1$, $g_2 = 1$ — $q_0 = x$
4. $g_1 = x + 0$, $g_2 = x$ — $q_0 = x$
5. $g_1 = x + x'y$, $g_2 = x + y$ — $q_0 = x, q_1 = y$
6. $g_1 = x + ab$, $g_2 = (x + a)(x + b)$ — $q_0 = a, q_1 = b, q_2 = x$
7. $g_1 = (xy)'$, $g_2 = x' + y'$ — $q_0 = x, q_1 = y$
8. $g_1 = (x + y)'$, $g_2 = x'y'$ — $q_0 = x, q_1 = y$

Figure 1. Equations to verify.

A. Parts List

- Clock
- 5V Source
- 74LS163
- Binary switch
- Resistor
- 2-input OR gates
- 2-input AND gates
- NOT gates

B. Truth Tables

State	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
q_0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
q_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
q_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
q_3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Figure 2. Truth table for a 74LS163 4-bit counter.

C. Schematics

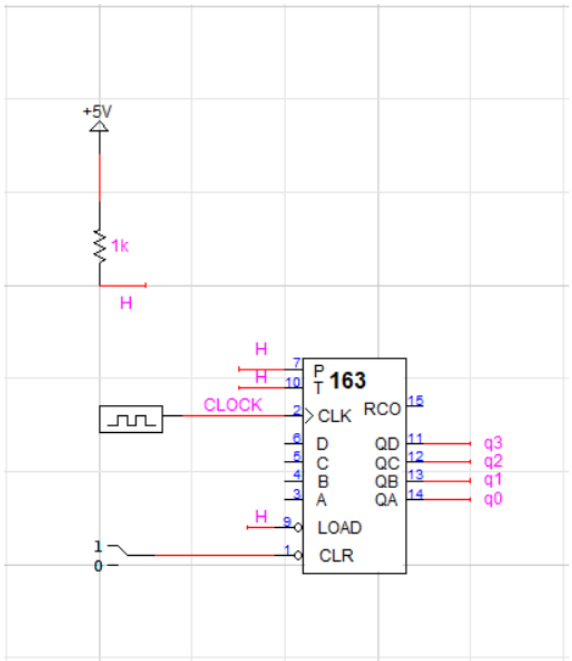

Figure 3. 74LS163 clock-counter circuit scheme.

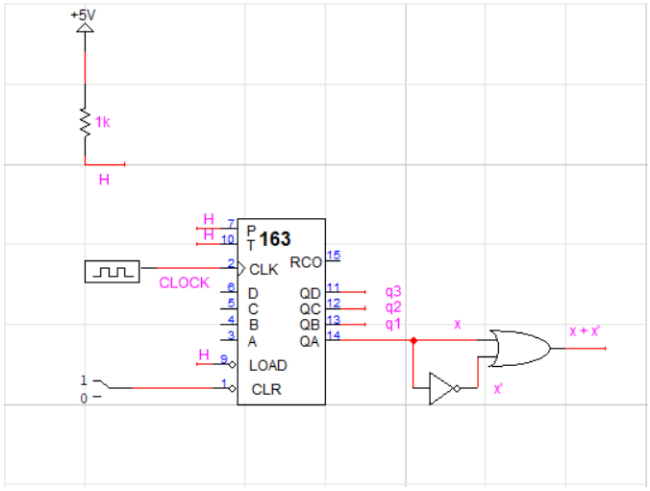

Figure 4. Circuit 1 for equation 1.

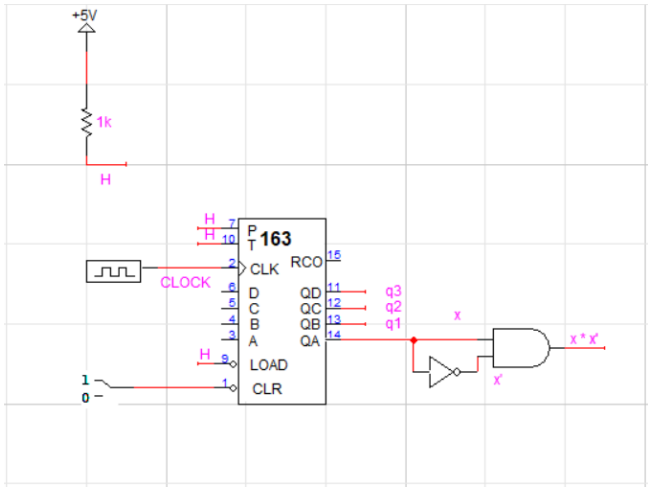

Figure 5. Circuit 2 for equation 2.

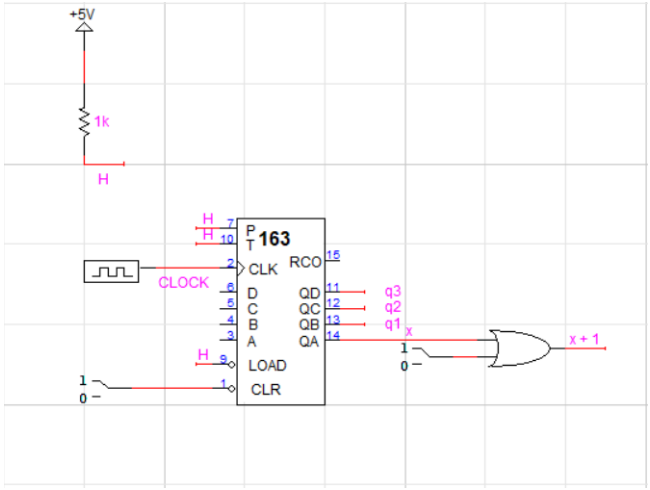

Figure 6. Circuit 3 for equation 3.

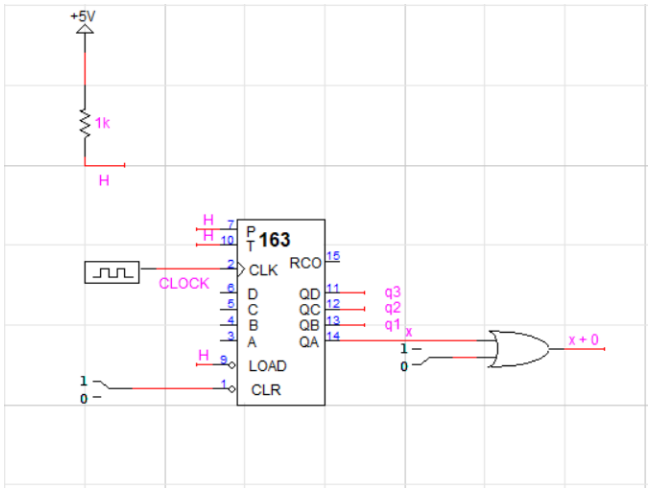

Figure 7. Circuit 4 for equation 4.

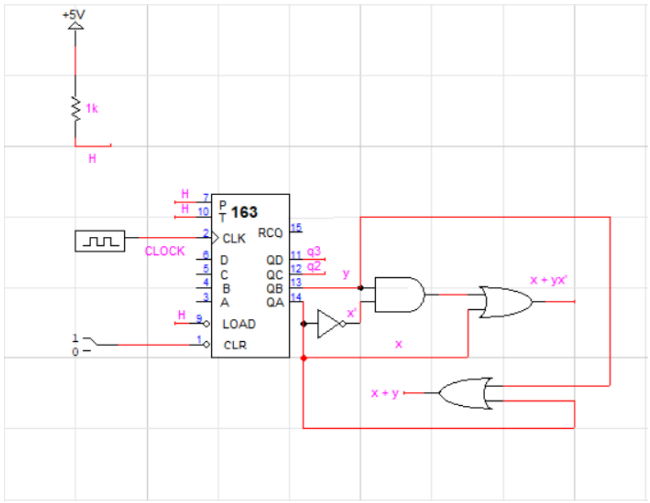

Figure 8. Circuit 5 for equation 5.

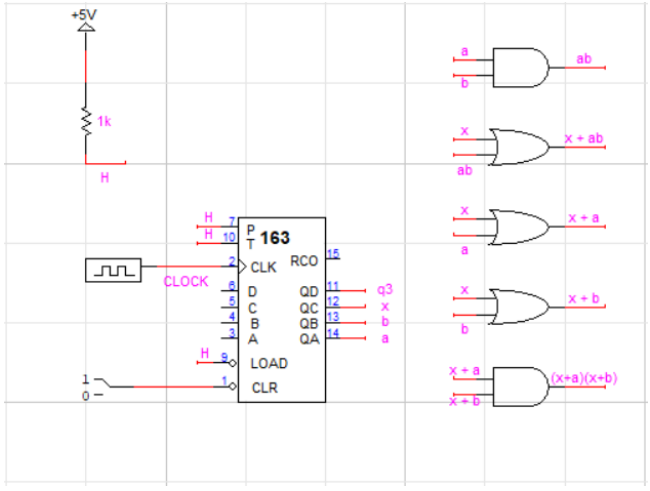

Figure 9. Circuit 6 for equation 6.

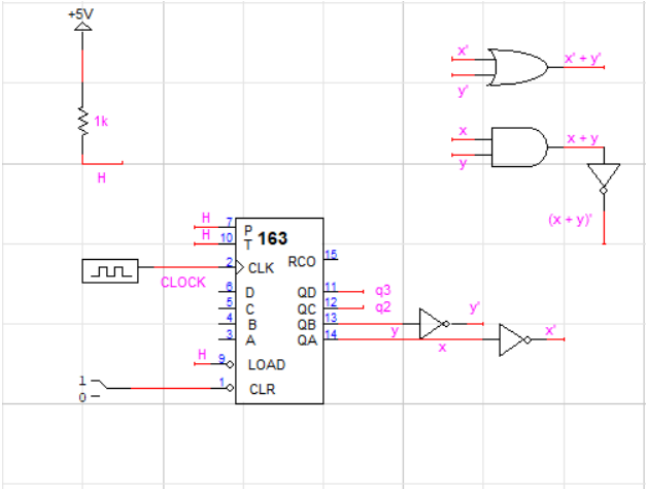

Figure 10. Circuit 7 for equation 7.

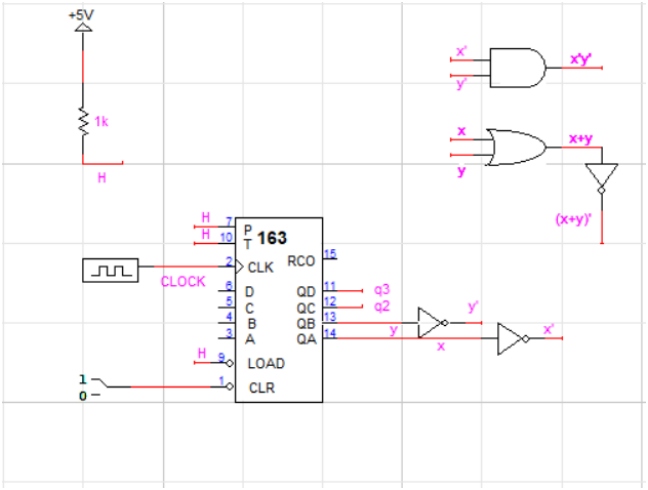

Figure 11. Circuit 8 for equation 8.

III. TESTING PROCEDURES

1. Create clock-counter circuit as pictured in Figure 3.
2. Reset the CLR with a switch.
3. Set up a circuit with OR, AND, and NOT gates to prove the previously given equations.
4. Run the clock and record the output for each setup.

IV. TESTING RESULTS

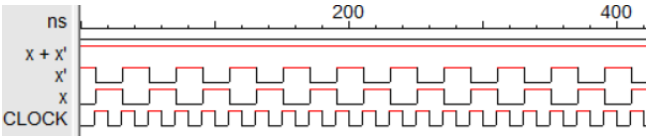

Figure 12. Circuit 1's simulation results.

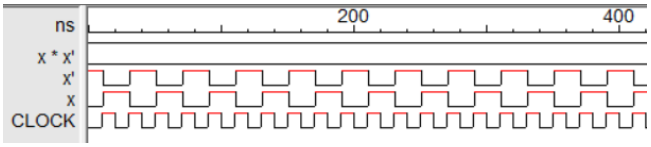

Figure 13. Circuit 2's simulation results.

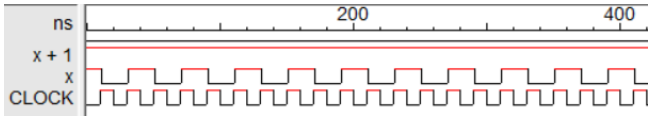

Figure 14. Circuit 3's simulation results.

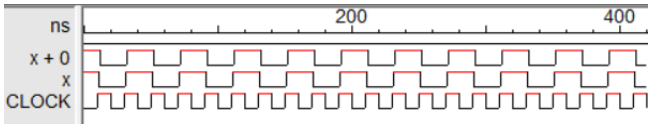

Figure 15. Circuit 4's simulation results.

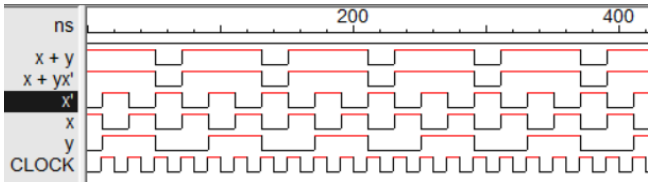

Figure 16. Circuit 5's simulation results.

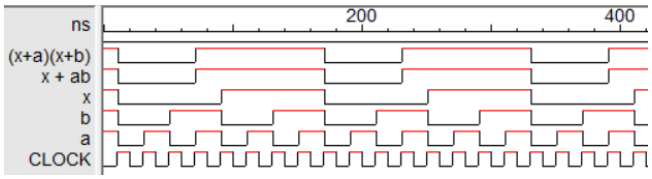

Figure 17. Circuit 6's simulation results.

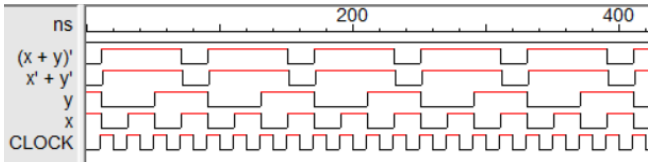

Figure 18. Circuit 7's simulation results.

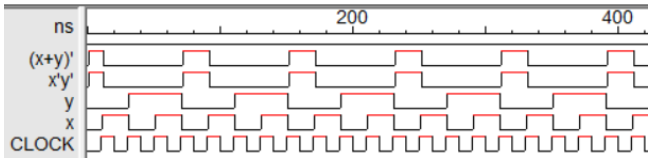

Figure 19. Circuit 8's simulation results.

V. CONCLUSION

In conclusion, a 4-bits binary number can be represented by a 74LS163 clock-counter circuit. The q0 output represents the lowest bit, changing and keeping that state as soon as the clock wave function changes, and q1 does the same with q0, as well as q2 with q1, and q3 with q2. The q0, q1, q2, and q3 wave functions encode for a 4-bit binary number that depends on the circuit's clock, hence we call our circuit a 4-bit clock-counter.

Furthermore, we can use the signals this clock outputs to prove the digital logic theorems we've been learning in our lectures, by incorporating AND, OR, and NOT gates.